UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/815,294	04/01/2004	Edwin Franklin Barry	800.0015 (A1145)	8328

73846 7590 08/19/2008

Peter H. Priest
5015 Southpark Drive, Suite 230
Durham, NC 27713

EXAMINER	
DOAN, DUC T	
ART UNIT	**PAPER NUMBER**
2188	
MAIL DATE	**DELIVERY MODE**
08/19/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS

AND INTERFERENCES

Application Number: 10/815,294

Filing Date: April 01, 2004

Appellant(s): BARRY ET AL.

Peter H. Priest #30210

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 6/16/2008 appealing from the Office action mailed 12/26/2007.

(1) Real Party in interest

A statement identifying by name the real party interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

US 6823505	Dowling, Eric M	11-2004

US 6944747	Nair et al.	09-2005
US 6173389	Pechanek et al.	01-2001
US Pub 2002/0199084	Choquette et al.	12-2002

Pentium Processor Family Developer's manual Volume 3:

Architecture and Programming Manual, Intel 1995

Attachment: TMS320 SECOND-GENERATION

DIGITAL SIGNAL PROCESSORS 1990

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject

matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-2, 4-6, and 16-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dowling (US 6823505) and in view of Intel Pentium Processor Family Developer's Manual Vol. 3, hereby Intel.

As in claim 1, Dowling discloses a processor address translation apparatus for translating an instruction operand address to a different operand address (Dowling's Fig 2, processor address translation apparatus comprises #102 register file, #150 etc. programmable address translation logic, for translating an instruction operand address, #Fig 2: #107, to a different address, see column 13, Table 5),

Dowling further discloses a memory with an address input for selecting a data element from a plurality of data element (Dowling's Fig 2: #AR0-ARn and associating auxiliary memory registers R1-RN, Dowling's column 1 lines 20-50),

Dowling's column 13, table 5 further discloses an instruction register for receiving an instruction encoded with an operand address (Table 5 instructions) and control information indicating the operand address is to be translated as part of the instruction's execution (Dowling's table 5, column 13 lines 19-35 further discloses by using the control information (i.e new "++" notation indicating address translation), the address translation apparatus translating the operand address into the translated address of the corresponding transpose element) ; and

an address translation unit for accessing the memory in a translation pattern, having the operand address as input and (Dowling's Fig 2,address translation apparatus comprising of # 102, #150 etc.. accessing the memory associating with #102 AR0-Arn in a translation pattern),

Dowling further discloses in response to the instruction received in the instruction register (Dowling's Table 5 instruction), translating the operand address to form the different operand address in accordance with the translation pattern (Dowling's table 5, column 13 lines **19-35** further discloses by using the control information (i.e new "++" notation indicating address translation), the address translation apparatus translating the operand address into the translated address of the corresponding transpose element), the different operand address accessing a data element from the memory through the address input (Dowling's Fig 2, the translated/different operand address accessing data element of memory associating with #AR0-ARN).

Dowling does not expressly disclose the claim's amended aspect of"...directly translating the operand bit field received as input .. "when doing the address translation. However, Intel discloses a typical general purpose processor with instruction having operand address expressed indirectly addressing format (Intel's page 25-6 [BX+DI]) or direct addressing format (Intel's page 25-6 disp 16). It would have been obvious to one of ordinary skill in the art at the time of invention to include operand direct addressing format as suggested by Intel in Dowling's system thus the address translation scheme can be used with direct addressing format and thereby further allowing Dowling's teaching to be used with other popular mainstream processor.

As in claim 2, Dowling further discloses several address translation functions supporting several translation patterns (Dowling's column 6 lines 47-66, programmable address translation functions supporting any desired address translation patterns such as +8, +4 etc; Dowling's column 7 lines 1-15, address translation functions for FFT algorithm, bit reversing functions or

any functions for matrix calculation. Dowling Fig 5 further discloses of any translation parameters can be stored and easily programmed for processing any desired translation patterns, using well known, programmable logic devices such as PLD etc..; see Dowling's column 9 lines 40-65, PR0-PR3 etc..; column 13 lines 50 to column 14); Dowling further discloses an input to select a translation pattern from the plurality if supported translation (Dowling's column 7 lines **1-3,** input that selects several addressing modes).

As in claim 4, Dowling Fig 4A, column 30-65 further discloses that the instruction is a block load instruction (i.e a typical matrix operation instruction in processor TMS320C2x).
As in claim 5, Dowling's Fig 5, in another embodiment, discloses that the processor address apparatus disposed within a plurality of instruction operand address paths (Fig 5: several instruction operand paths from dispatcher units Fig 5 #606) for a plurality of instruction, the plurality of instructions fetched for simultaneous execution (Fig 5:#617, #619 several programmable address translation functions are receiving plurality of instruction's operand addresses being fetched simultaneously).

As in claim 6, Dowling further discloses the plurality of instructions constitute a very long instruction word (VLIW) (Dowling's columns 50-56 further disclose the programmable address translation functions are incorporated into different functional and/or executing units executing instructions that constitute the VLIW instruction).

As in claim 16, Dowling discloses a processor address translation method for translating an instruction operand address to a different operand address, the address translation method comprising: receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution; translating

the operand address according to a function; and accessing a data element with the translated address (see rationale of claim 1), and repeating the receiving, translating, and accessing steps to access data elements in a pattern according to the function (Dowling's table 5, repeating steps for sequence of inputs according the function auto incrementing).

As in claim 17, Dowling further discloses wherein the function comprises combinatorial logic for translating the operand address (Dowling's column 9 lines 1-7, comprises combinational logic).

Claims 3 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dowling (US 6823505), Intel as applied to claims 2 and 16 respectively and in view of Nair et al (US 6944747).

As in claim 3, Dowling column 9, lines 40-65 discloses using the parameter registers PR0-PR3.. for providing translated addressed for automatic indexing FFT, and matrix operations (see Dowling's column 9 lines 5-15). Dowling does not expressly disclose of providing the bit complementing function for matrix operation (corresponding to the claim's e bits function). However, Nair teaches an apparatus Fig 1 to process several matrix operations (Nair's table 1 and 2), including providing bit complementing function for matrix operations, Nair's column 12 lines 13-30, bit-wise complement operation, bit-wise complement operation corresponding the claim's matrix operation with e bit). It Would have been obvious to one of ordinary skill in the art at the time of invention to include bit-wise complementing function and matrix operations as suggested by Nair into Dowling's system modified by Intel, thereby the system can further providing additional complex matrix operations in an automatic manner as shown in Nair's

Table 1, and 2, and thereby further improve the overall throughput in the system for executing matrix operations.

Claim 19 is rejected based on the same rationale as of claim 3.

Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dowling (US 6823505) and in view of Nair et al (US 6944747).

As in claim 18, Dowling discloses an address translation method for translating a first address of a first data element in a memory to a second address of a second data element in the memory (Dowling's operand address translation circuitry to translated a first address into the second address for matrix operation, see Dowling's column 7 lines 1-15, column 6 lines 47-67), the address translation method comprising:

Dowling discloses enabling an address translation unit for translation (Dowling's column 7 lines 1-15, selecting and enabling the addressing mode for a particular address translation pattern FFT, matrix operation etc..) initiating a read operation to read a first data element at a first address during a read operation (It's is noted that any FFT, matrix operation requires reading a first source data element); translating the first address to the second address in accordance with the {s, e} bit specified translation pattern (Dowling's Fig 2,operand address translation function #212 translating address accordance with the algorithm pattern, see Dowling's column 7 lines 1-15) ; and completing the read operation by reading the second data element at the second address (Dowling's Fig 2, using the translated operand address to read the second data element that is the result of an FFT bit reserving addressing step).

Dowling does not expressly disclose the claim's detail of matrix operation with the bit complementing function. However, Nair teaches an apparatus Fig 1 to process several matrix

operations (Nair's table 1 and 2), including providing bit complementing function for matrix operations, Nair's column 12 lines 13-30, bit-wise complement operation (corresponding the claim's matrix operation with e bit). It would have been obvious to one of ordinary skill in the art at the time of invention to include bit-wise complementing function and matrix operations as suggested by Nair into Dowling's system, such that the system can further provide additional complex matrix operations as shown in Nair's Table 1, and 2 in an automatic manner, and thereby further improve the overall throughput in the system for executing matrix operations. Nair further discloses typical matrix operation comprises of determining a set of {s, e} bits that specify a translation pattern; loading the set of {s, e} bits into an address translation parameter control register (Nair's loading the matrix parameter set of {s,e} representing by the mnemonic as shown in table 2); Dowling further discloses that any parameter set can be easily stored for operations such as matrix, FFT etc.., using well known programmable logic device PLD, see Dowling's column 14 lines 50 to column 14).

It's noted that the amended claim 18 merely rearranges some components of the address translation logic such that some of these elements are now claimed be located in "the address translation memory". Clearly, simply arranges old elements with each performing the same function it had been known to perform and yields no more than one would expect from such an arrangement, the combination would be obvious. In this instant, Dowling's Fig 2 teaches each element of the claim's "address translation memory", and they each performing the same function it has been known to perform and yields no more than one would expect from such an arrangement (address registers 102 are used for storing addresses being translated. Data memory 120 to store corresponding data pointed to by the address in the address registers 102. Address

translation is disclosed by logic in 150, that including logic for parameter control logic (Dowling's Fig 3A: PR3,PR1 that control address translation from input address registers to output address registers), thus the combination would be obvious.

Claims 7-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pechanek et al (US 6173389) and in view of Dowling (US 6823505).

As in claim 7, Pechanek discloses typically a computer system, for example a VLIW processor system, Fig 3, comprises of register file and associating register file addressing/indexing logic (i.e. corresponding to the claim's RFI register file indexing logic and updating logic) to store operands in register file such that operands being stored in the register file can be quickly accessed by instruction in functional units of the processor (i.e. load and store registers of register file for executing instruction such as add and multiply, see Pechanek column 1 lines 47-61). Pechanek does not expressly disclose the logic to translate the operand addresses of sequence of instruction to an RIF sequence of different operand addresses. However, Dowling discloses an programmable address translation apparatus capable of translating a sequence of instruction operand addresses into different operand addresses of sequence of instructions (Dowling's Fig 2, column 6 lines 47-67, address register file AR0-ARn register file # 102, column 13 Table 5) for several addressing translating pattern (i.e addressing modes to translate addresses of FFT patterns, and/or addressing patterns corresponding to matrices manipulation). It would have been obvious to one of ordinary skill in the art at the time of invention to include translation operand addresses into different operand addresses as suggested by Dowling such that instructions can be completed faster and thereby further improve the overall system's throughput (see Dowling's column 13 lines 19-50).

Dowling further discloses a memory with an address input for selecting a data element from a plurality of data element (Dowling's Fig 2: #AR0-ARn and associating auxiliary memory registers R1-RN, Dowling's column 1 lines 20-50),

Dowling's column 13, table 5 further discloses an instruction register for receiving an instruction encoded with an operand address (Table 5 instructions) and control information indicating the operand address is to be translated as part of the instruction's execution (Dowling's table 5, column 13 lines 19-35 further discloses by using the control information (i.e new "++" notation indicating address translation), the address translation apparatus translating the operand address into the translated address of the corresponding transpose element) ;

Although Pechanek Fig 3 discloses a VLIW processor system having RFI, register file and associating updating logic, Pechanek does not expressly discloses the claim's detail of the associating logic to manipulate operand addresses. However, Dowling further discloses a logic that manipulates operand addresses (i.e manipulating operand addresses corresponding to address register files, Dowling's Fig 2: #AR0-ARn and associating auxiliary memory registers R1-RN, Dowling's column 1 lines 20-50), capable of generate on the register file update unit's output a linear sequence of register file operand addresses in response to received sequence of register file translation type instruction (Dowling's column 2 lines 40-50, translating operand addresses using linear auto incrementing mode).

Dowling further discloses a multiplexer for selecting between the operand address from the instruction register for a first register file operation and subsequence register file operations (Dowling's Fig 2: for selecting not translated the operand address, therefore address from the instruction register # 107 or translated operand address from the programmable address

translation circuitry #212. It's further note that for incrementing addresses pattern, the first address is not being translated, and the subsequent addresses are translated to +8, -4 etc.. see Dowling's column 2 lines 50-58, thus a multiplexer is required for bypassing/not using the address translation function #217; Dowling's column 9 lines 1-15, teaches by providing feedback path of translated operand addresses (i.e output of translation function #217) to the registers #102, using for example multiplexer circuitry #203, to implement sequential logic functions. In other words, Dowling clearly teaches of using multiplexer to provide addresses not being translated, and to provide translated addresses for subsequent cycles);

Dowling further discloses an address translation unit for access the memory in a translation pattern, receiving a sequence of operand address from the multiplexer (Fig 2:#203 multiplexer feeding back path of translating addresses to the registers #102 and proving them to address translation function #212 in subsequent cycles) and in response to the sequence of RFI operand addresses, translating the sequence of RFI operand addresses to form a sequence of different operand addresses (Dowling's column 9 lines 1-15, teaches by providing feedback path of translated operand addresses (i.e output of translation function #217) to the registers # 102, using for example multiplexer circuitry #203, to implement sequential logic functions. In other words, Dowling clearly teaches of using multiplexer to provide addresses not being translated, and to provide translated addresses for subsequent cycles); in accordance with the translation pattern (Dowling's table 5, column 13 lines 19-35 further discloses by using the control information (i.e new "++" notation indicating address translation), the address translation apparatus translating the operand address into the translated address of the corresponding transpose element), the different operand addresses each accessing a data element from the

memory through the address input (Dowling's Fig 2, the translated/different operand address accessing data element of memory #AR0-Arn).

As in claim 8, Dowling's Fig 5 further discloses the RFI address translation apparatus disposed within PEs of an array of PEs (Dowling: Fig 6A array of PEs comprise address translation apparatus representing by #617, #619).

As in claim 9, Dowling's Fig 6A further discloses the processor RFI address translation apparatus of claim 7 disposed within a plurality of instruction operand address paths for a plurality of instructions, the plurality of instructions fetched for simultaneous execution(Fig 6A).

As in claim 10, Dowling further discloses the processor RFI address translation apparatus of claim 9 wherein the plurality of instructions constitute a very long instruction word (VLIW) (Dowling's column 50-56 further discloses the programmable address translation functions are incorporated into different functional and/or executing units that executing instructions that constitute the VLIW instruction).

Claims 11-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dowling (US 6823505) and in view of Choquette et al (US 2002/0199084).

As in claim 11, Dowling discloses an address translation memory device for accessing data at translated addresses, the address translation memory device comprising (Dowling's Fig 2, processor address translation apparatus comprises # 102 register file, # 150 etc. programmable address translation logic, for translating an instruction #Fig 2: #' 107 operand address to a different address, see column 13, Table 5): a first read address input (Dowling's Fig 2: a first read input for reading operand address from instruction Fig 2:#107);

Although Dowling Fig 2, column 1 lines 40-45 discloses a register file structures comprises of address registers Fig 2:#AR0 that provide addresses to associating operand data stored in memory/auxiliary registers R1-Rn (corresponding to the claim's storage device). Dowling does not expressly disclose the claim's aspect of ports associating with the storage device. However, Choquette discloses a typical computer system having several execution units that operate parallel and using a register file structure Fig 1 : #30. In such a system, the register file structure Fig 1:#30 (corresponding to the claim's storage device) capable of providing several read, write address ports, and several data in data out ports so that several execution units can access the file register concurrently (see Choquette paragraph 4). It would have been obvious to one of ordinary skill in the art at the time of invention to include the register file structure with several read, write address ports and several data in data out ports so that several execution units can executing and accessing the file register structure concurrently, and thereby improve the overall system's throughput (see Choquette, paragraphs 1, 4).

Dowling further discloses the operand address translation apparatus for accessing the storage device in a translation pattern, the address translation unit translating the first read address input in accordance with the translation pattern (Dowling's Fig 2: # 3217 programmable address translation function translates address in accordance with translation pattern, auto increment or FFT etc.., see Dowling's column 6 lines 45 to column 7 line 15), to the storage device second read address input, for reading data from the storage device at a translated address during read operation (Again, it's further noted as disclosed by Choquette in above paragraph, several storage ports (read write addresses data ports are provided so that the register file can be accessed by several executing units concurrently).

It's noted that the amended claim 11 merely rearranges some components of the address translation logic such that some of these elements are now claimed be located in "the address translation memory". Clearly, simply arranges old elements with each performing the same function it had been known to perform and yields no more than one would expect from such an arrangement, the combination would be obvious. In this instant, the recited references teach each elements of the claim's "address translation memory device", and they each performing the same function it has been known to perform and yields no more than one would expect from such an arrangement (address registers 102 are used for storing addresses being translated. Data memory 120 to store corresponding data pointed to by the address in the address registers 102. Address translation is disclosed by logic in 150, that including logic for parameter control logic (Dowling's Fig 3A: PR3,PR1 that control address translation from input address registers to output address registers), thus the combination would be obvious.

Claim 12 discloses a variation of claim 11, and directs to a write operation. The claim is rejected based on the same rationale as of claim 11. Again, it's further noted as disclosed by rationale in claim 11, several storage ports, read, write, addresses, and data ports are provided so that the register file can be accessed in either read or write operations by several executing units in a concurrently manner.

As in claim 13, Dowling's Fig 2 clearly discloses processor address translation apparatus comprises #102 register file, #150 etc. programmable address translation logic, for translating an instruction #Fig 2: # 107 operand address to a different address, see column 13, Table 5; Dowling further discloses wherein the storage device (auxiliary register R0-Rn) having location selection logic (Fig 2: #AR0-AR) merging with the address translation unit (Fig 2:#217).

Claim 14 is rejected based on the same rationale as of claim 2.

Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dowling (US 6823505), Choquette et al (US 2002/0199084) as applied to claim 14, and further in view of Nair et al (US 6944747).

As in claim 15, none of Dowling and Choquette discloses the claim's aspect of complementing bit function of matrix operation. However Nair teaches the claim's limitation as discussed in the rationale of claim 3.

(10) Response to Argument

Appellant's arguments in their brief have been fully considered but they are not persuasive. Examiner respectfully traverses Appellant's arguments for the following reasons:

A) Appellant's arguments regarding the rejections of claims 1-19 under 35 U.S.C 103(a) are not persuasive.

A0) The instant application as a whole discloses a technique to minimize the instructions being processed by the computer to execute a sequence of steps on a set of elements. Typically, the steps are comprised of:

Step 1. Load the source elements,

execute a function on the source elements.

store the result to destination elements.

Step 2. When the results of the first step are used in the second step, in certain applications such as FFT,

transferring the result of destination elements to source elements of step 2 (1)

Load source elements,

Executing a function on the source elements.

store the result to destination elements.

Recognizing that source elements in the second step are the same as the result of the previous step. It can be seen that by translating/transforming the addresses of the source elements in the second step to point to the destination element of the first step, operation transferring result of destination elements to source elements of step 2 (1) can easily be eliminated. Thus the overall operation will be executed faster and more efficiently.

Dowling teaches the above concept same as being disclosed in the instant application, that is taking the advantage of translating/transforming the addresses of the source elements in the second step to point to the results of the first step stored in Fig 1: AR0-ARn, see Dowling's col. 2 lines 1-40.

Additionally, Examiner notes that Appellant's argument dated 10/15/2007, pages 11-12 admitted that Dowling teaches the addressed translation/transformation, "Dowling addresses programmable address modes for calculating effective addresses based on address values stored in address registers for operands stored in a data memory". However, Appellant argues that Dowling's addresses translations directs to these address registers and "does not "directly" translate the operand address bit field received as input to form the different operand address with the different operand address accessing a data element from the memory through the address inputs as claimed in amended claim 1.

In other words, Appellant amended the claims on 10/15/2007 to distinguish from Dowling's addresses translation/transformation with operand bit field that point directly to the memory location. Support for this amendment can be seen in specification's Fig 3 in which Rt

320 represent the address register that is similar to Dowling's AR0-ARn, and Fig 3 325 14-bit direct address represents the claim's amended directly translated operand bit field.

Dowling does not expressly disclose this variation of operand address (directly point to the memory location). However Intel clearly teaches that in a computer system, there are several known variation of operand addresses which can be seen organized to point to a set of registers or point to memory location directly (memory displacement value). In Intel, the displacement value points to a memory location. Therefore the Intel's displacement value is type b operand address as claimed. It would have been obvious to one of ordinary skill in the art at the time of invention to include operand direct addressing format as suggested by Intel in Dowling's system thus the address translation scheme can be used with direct addressing format and thereby further allowing Dowling's teaching to be used in several types of computer systems.

Therefore Appellant's argument is not persuasive.

With regard to the rejection of claims 1, 2, 4-6, 16 and 17,

A1) Per claim 1, Appellant argues "..Dowling does not transform this bit field encoded with the selected address register, but rather operates to transform the contents of the selected address register, as noted above with regard to Dowling's Figs. 3A and 3C… The Examiner correctly admits that Dowling does not expressly disclose "directly translating the operand (address) bit field received as input..." when doing the address translation. The Examiner suggests that Intel discloses a direct addressing format and cites Intel page 25-6 displ6. However, at Intel page 25-6 Notes: 2 "displ 6 denotes a 16-bit displacement following the ModR/M bytes, to be added to the index." As described in Intel, displ6 is not a direct addressing format but, rather, is used in the generation of an effective address by adding the displ6 value to an index

value. Intel does not teach and does not make obvious an the displ6 value to an index value. Intel does not teach and does not make obvious an "instruction register for receiving an instruction encoded with an operand address in an operand address bit field of the instruction and control information indicating the operand address is to be translated as part of the instruction's execution" as claimed in claim 1.."

Examiner disagrees. The instant application discloses an address translation scheme to translate operand address of two types a) the operand field of an instruction is an address/logical number **that point to a register** of registers. This register number is regarded in the instant application as indirect type address and b) the operand field of an instruction refers **directly to address/pointer to memory location.** This pointer value in the operand field is viewed in the instant application as direct type address and recited as "an operand address bit field of the instruction" in claim 1. Notes that the address translation scheme is applied readily and equally well to register (using type a address) and/or memory location (using type b address). Dowling discloses the address translation scheme of the instant application direct to the operand address of type a as an example. Dowling does not expressly disclose the operand address as type b. However, Intel teaches that typically the operand address field of an instruction can be several variations, including type b address of the instant disclosure. In Intel, the displacement value points to a memory location. Therefore the Intel's displacement value is type b operand address as claimed. It would have been obvious to one of ordinary skill in the art at the time of invention to include operand direct addressing format as suggested by Intel in Dowling's system thus the address translation scheme can be used with direct addressing format and thereby further allowing Dowling's teaching to be used in several types of computer systems.

Appellant further argues, "The Examiner further states that it would have been obvious to one of ordinary skill in the art at the time of invention to include an operand direct addressing format as suggested by Intel in Dowling's system. However, Dowling does not show **a direct addressing path from an instruction register** for receiving an instruction with an operand address and control information as claimed in claim 1. It is also not clear how Dowling could provide such a path since Dowling does not mention an instruction register. Further, Dowling provides no motivation that such a path would provide any value. Rather, Dowling states that the programmable AAU 301 may be used "as the programmable AAU 212 in the processor 200 shown in Fig. 2" which "**allows a programmer to programmable permute the bits in the address registers AR0-ARn in the register set 102**."' Dowling, in Figs. 2, 3A, and 3C and col. 10, lines 14-18. Dowling merely extends the capabilities of an address arithmetic unit (AAU) by programmable modifying address values stored in address registers. Dowling does not teach and does not make obvious an "instruction register for receiving an instruction encoded with an operand address in an operand address bit field of the instruction and control information indicating the operand address is to be translated as part of the instruction's execution" as claimed in claim 1.

In responses, Appellant argument is not understood and clearly not commensurate with the claim language. Examiner cannot find any statement in claim 1 that claims “a direct addressing path from an instruction register”. Thus the argument is not relevant. Moreover, an instruction register is inherently required for processing instruction in a processor, see Dowling’s Fig 2. Additionally, Appellant admits that Dowling teaches manipulate address bit of operand address registers AR0-ARn, but somehow does not teach the claim’s operand address

translation. Examiner disagrees, Dowling's manipulation of operand address bits AR0-ARn covers the claim's operand address translation (the AR0-ARn in the register set 102 stores the operand address, col. 1 lines 20-51; col. 9 lines 1-15, col. 10 lines 14-18, by manipulating the address bit by the functions of a logic device AA212, the operand address translation is achieved, "**the programmable AAU 212 can be programmed to provide automatic address indexing for FFT processing, Viterbi decoding, discrete cosine transform, circular buffers etc**..".

Therefore, the argument is not persuasive.

A2) Per claim 4, Appellant argues 'Regarding claim 4, the Examiner suggests that Dowling's Fig. 4A and column 10, lines 30-65 disclose a block load instruction. The Official Action further suggests that a block load instruction is a typical matrix operation instruction in processor TMS320C2x. However, Dowling's Fig. 4A merely illustrates a matrix having sixteen data elements. The text at Dowling's col. 10, lines 30-60 does not indicate a block load instruction, but merely indicates that the data elements of the matrix 400 shown in Fig. 4A are "actually stored in a linear array in memory" and as "shown in FIG. 4B, in memory, the sixteen elements of the matrix 400 are laid out sequentially in memory". Emphasis added. Dowling, col. 10, lines 58-60 and lines 34 and 35. Dowling makes no mention of a singular instruction which may load a block of data as claimed. Dowling does not teach and does not make obvious a block load instruction as claimed..".

In response, Dowling teaches that the address translation can be applied to matrix operation executed by processor such as TMS320C2x (Dowling's Fig 4a, col. 10 lines 30-65 and col. 1 lines 40-45). Typically, a processor, for example TMS320C2x processor, further provides a block instruction to process efficiently blocks of data , for example blocks of data used in a

matrix operation (The block instruction BLKD is discloses in Table 3 I/O and data memory operation of the TMS320 SECOND-GENERATION DIGITAL SIGNAL PROCESSORS document that is shown here as evidentiary reference). Thus Examiner submits that block instruction is a well known instruction and used to further process blocks of data quickly and efficiently. And one skill in the art at the time of invention would recognize the efficiency of the block instruction and applying the instruction to move data blocks, for example data blocks of matrix operation, in an efficiently manner and thereby further improve the performance of the overall system.

Per claim 16, Appellant offers the same argument as of claim 1 and the same response apply. As such, the argument is found to be not persuasive.

A3) Per claims 3 and 19, Appellant argues "…The Examiner focuses on one particular aspect of claim 3 concerning the claim's "e" bits function. The Examiner suggests that Nair teaches a bit complementing function that may be applied to an operand address bit field. However, Nair merely describes operations on data elements and not address bits. Also, Nair does not teach and does not make obvious translation parameters that "include k by k s bits and k e bits for a k bit address" as claimed in claim 3. Further, Nair does not teach and does not make obvious "combinatorial logic governed by the following equations, where the operand address bit field input is A0, A1 .....A(k-1), product operations are treated as ANDs, sum operations are treated as XORs, and the different operand address is A0', AI', ...A(k-l)'..".

In response, Appellant argument seems to be describing the inherent characteristics of the matrix transformation having a parameter matrix {S,E} grouping as two parts to show the distinction of

a_group) S = k by ks bits matrix that represents other transformation functions, and

b_group) E = k by s bits matrix that represent parameter and the function to inverting certain address bit. Examiner notes the ebits are bitwise ANDing with logical 1.

The basic bitwise operation of binary bit is well known in the art, as taught by Nair at col. 12 lines 13-31, " In addition to the illustrative arithmetic operations described in table 1, basic bit-wise logical operations including bit-shifting, **logical AND/NAND/OR/NOR/XOR and similar logical operations known to those skill in the art may also be performed on the various matrix elements**. …In some case **the bitwise complement operation**, there may actually be only two operand matrices with the results from the operation on a source matrix being stored in a destination matrix.." . Thus Nair clearly teaches that elements/bits of matrix can be transformed using bitwise operations. Such as, to use a bit A, it is known and is represented as logical bitwise AND operation with 1, A = A AND 1. Not to use a bit A, it is known and is represented as logical AND operation with a zero 0, simply because regardless of the value of the source bit A (one or zero) when it is ANDing with zero, the result is zero, 0 = A AND 0. **To inverting value of a bit A, is known and is represented as logical bitwise XOR operation with a logical 1, that is A invert = A XOR 1**. And consequently when exclusive with zero, it is known to give the same result A = A XOR 0.

Thus, Examiner submits that the matrix as recited in claim 3 grouped a group and b group above, merely indicate the basic bitwise operations as taught by Nair above, such as XORing A with logic 1 to invert a bit A, using and not using a bit A by ANDing A with logic 1 or logic 0. Examiner further submits that to invert/complement a bit A, **the bit wise XOR** with logic 1 **must be used**.

Finally, the basic bitwise operations as taught by Nair can benefit for matrices operations equally well for translating of any binary bit matrices such as data, code, address, instruction etc.. Therefore Nair's teaching of bit wise operations including bit complementing function is readily to be applied in Dowling's address translation scheme and including bits represent as "operand address bit field input" in Dowling's system.

And therefore Appellant's argument is not persuasive.

A4) Per claim 18, Appellant argues that Dowling does not teach the claim limitation "address translation method for translating a first address of a first data element in an address translation memory".

First, Examiner cannot find anywhere in the specification that clearly defined and set forth the "address translation memory" as claimed. In fact the phrase "address translation memory" is not found anywhere in the specification. Appellant's brief page 13 second paragraph which supposes to map the claim's language of the above limitation, states " See, for example, Figs. 8A, 9, 10A, and 10B, page 21, line 10- page 22, line 1, and page 24, line 10 - page 26, line 5 which describe translation operations including a transpose operation ". However, these paragraphs discuss translation operations using slightly modified arraignments of known elements/components in the translation operations, and still the phrase "address translation memory" is not defined/found anywhere. Therefore, 'the address translation memory" can only reasonably interpreted as a memory article/device with some logic and some storage units to store bits using in the translation operation. The details of storage units to store these bits are not important so long as they store bits being used in translation operations. In fact, Examiner notes that Appellant deliberately uses various terminologies to describe the **same** storage element/bits,

such as register file or memory unit (specification's page 21 line 11); And page 22 lines 1, "storage unit 835" and line 17, "storage device 835" clearly suggests the interchangeable of the terminologies "storage unit" and "storage device".

Second, specification page 4 states "Fig. 8A **illustrates an exemplary register file or memory unit** incorporating the address translation function and translation parameter state internally with a view of a read port in accordance with the present invention" ; Specification's page 21 lines 10-14 further states "Another aspect of the present invention, is achieved by **including the address translator in the register file or memory unit** thereby creating a storage unit a programmer would view as a pool of register or memory locations that can be manipulated by operations on storage unit addresses". Therefore, the specification clearly suggests a) the storage bits can be register files and the register files can be regarded as a memory unit. b) The main different of Fig 8A is and Fig 2D is the re-arraignment by putting the address translation function and memory storage elements together and labeling it "an address translation memory". Clearly, simply arranges old elements with each performing the same function it had been known to perform and yields no more than one would expect from such an arrangement, the combination would be obvious. In this instant, Dowling's Fig 2 teaches each elements of the claim's "address translation memory", and they each performing the same function it has been known to perform and yields no more than one would expect from such an arrangement (address registers 102 are used for storing addresses being translated. Data memory 120 to store corresponding data pointed to by the address in the address registers 102. Address translation is discloses by logic in 150, that including logic for parameter control logic (Dowling's Fig 3A:

PR3, PR1 that control address translation from input address registers to output address registers) thus the combination would be obvious.

Appellant further argues "..However, as noted in the specification at page 21, lines 10-15, the address translator is included in a register file or memory unit such as shown in Fig. 8A. In particular, "[c]consider an exemplary storage subsystem 800 of Fig. 8A, illustrating aspects of a read port, Which operates differently than the storage subsystem 238 shown in Fig. 2D". Also, as noted in the specification at page 21, line 18-.page 22, line 1, in "Fig. 2D subsystem 230, the address translation occurs only in operation for instructions that specify the address translation function. Other instructions, which don't specify an address translation function, use the register file or memory unit normally as sequentially addressed storage. **In the storage subsystem 800 of Fig. 8A, all address inputs 815 are translated** according to the translation settings 820 that govern how the addresses access data from the storage unit 835". Claim 18 does not claim a rearrangement of old elements providing an already known function".

In response, Examiner disagrees. Fig 2D clearly shows all address inputs must go through the address translation function 243, these address inputs are translated according to the load translation parameters 248 that govern how the addresses access data from the storage unit 238. In other words, Fig 2D behaves the same as Fig 8A, and all address bits are being translated. Additionally, it's noted, in both Fig 2D and Fig 8A, the translation means and can be interpreted as the inputs are changed or the inputs remain the same, see item A3 above.

The different between Fig 2D and Fig 8A wholly is the re-arrangement of components. For example the Fig 8A has one square box 810 wherein the Fig 2D has two square boxes 232 and 238. By putting the elements close to each other, the output signals from address translation

unit can quickly arrived at the l of 8 selector 241 sooner naturally, and therefore latching/port address register 239 is not required.

Third, Appellant argues "..Due to this error, it is not clear what the Examiner's is arguing. Yet it is due to this error that the Examiner dismisses the Appellants' remaining arguments as not being relevant. **Dowling's data memory 120 is not an "address translation memory" having an "address translation parameter control register" or having an "address translation unit" as claimed in** claim 18".

In response, Per last Office Action, dated 12/26/2007, page 17, Examiner states "As to Appellant's arguments regarding the rejection of the amended claim 18, Appellant argues that "Dowling's data memory 102 is not an address translation memory". The argument is not understood. Dowling's data memory 102 have the data correspond to **the address registers AR0 to ARn. These address registers contains addresses that used for address translation, and thus belong to "the address translation memory".** Appellant's remaining arguments are not relevant because they are based on the false assumption that Dowling's data memory is the claim's address translation memory".

Examiner was not clear why in Appellant's remarks dated 10/15/2007 page 14 to 15, Appellant argued about Data memory 120 that in Appellant's own words, "Dowling's data memory 120 is not an address translation memory".

Thus in the last office Action, Examiner clearly set forth that the AR0-ARn can be regarded as the address translation memory. In the Appeal Brief, Appellant now agrees that Dowling's data memory 120 is not an address translation memory". Thus it renders Appellant's

previous arguments, remark dated 10/15/2007 pages 14 to 15 which attempted to compare Dowling's data memory to address translation memory irrelevant.

Fourth, Appellant argues that Dowling does not teach "... As claimed in claim 18, with the "address translation parameter control register located in the address translation memory" and the "address translation unit located in the address translation memory", a read operation may be initiated "to read a first data element at a first address during a read operation", "translating the first address to the second address", and "completing the read operation by reading the second data element a the second address" as claimed in claim 18 ". Examiner disagrees.

In response, Dowling teaches,

"address translation parameter control register located in the address translation memory" (Fig 3A shows program registers 308 that map inputs 302 to output 304; the registers 308 in the AAU that can be viewed as the address translation memory, See the discussion in item A4 above. Notes the input 302 and output 304 from/to the register set 102, col. 8 lines 51-63);

"address translation unit located in the address translation memory" (Fig 3A the cross bar logic translates input 302 to output 304 is a part of AAU which can be viewed as the address translation memory, See the discussion in item A4 above);

"the read operation may be initiated to read a first data element at a first address during a read operation" (it's noted that an FFT or matrix operation using the address translation as taught by Dowling's Fig 3A, requires reading a first/source register 302 from the register file AR0-ARn. Thus for example an FFT read operation has a first reading to read a first element to 302),

"translating the first address to the second address" Fig 3A the cross bar logic translates input 302 to 301),

"completing the read operation by reading the second data element at the second address" (FFT operation reading continues by a second reading of the second element from register set 102. See col. 7 lines 1-15, col. 9 lines 1-15).

Appellant further argues "Dowling translates an address before any read operation to the data memory 120 occurs".

In response, Appellant argument is not understood. As discussed in above paragraphs, Dowling clearly teaches the sequences of reads as recited in the claim.

Appellant further argues "As shown in Dowling's Fig. 3A, an input from address register 302 is permuted to an output to address register 304. As also shown in Dowling's Fig. 2, the output of programmable AAU 212 is selected by multiplex 203 and provided to the register set 102 comprising address registers. **It is the output of the address registers which is selected by multiplex 122 to address Dowling's data memory 120**" .

In response, Again, Appellant refers to the irrelevant Dowling's data memory 120 that was discussed above.

Thus Dowling teaches all of the claim's 18 features as recited. And therefore Appellant's argument is not persuasive.

A5) Per claims 7-10,

Appellant argues, "Claims 7-10 were rejected under 35 U.S.C. §103(a) based On Pechanek 6,173,389 in view of Dowling. The Examiner admits that Pechanek 6,173,389 does not expressly disclose the logic to translate the operand addresses of a sequence of instructions to an RFI sequence of different operand addresses and depends upon Dowling to resolve the admitted deficiency. As noted above, Dowling does not teach and does not make obvious an "instruction

register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution" as claimed in claim 7. Thus, Dowling does not resolve the admitted deficiency of Pechanek 6,173,389".

In response, Dowling's column 13, table 5 further discloses an instruction register for receiving an instruction encoded with an operand address (Table 5 instructions) and control information indicating the operand address is to be translated as part of the instruction's execution (Dowling's table 5, column 13 lines 19-35 further discloses by using the control information (i.e new "++" notation indicating address translation), the address translation apparatus translating the operand address into the translated address of the corresponding transpose element).

Appellant further argues, "The Examiner suggests that Dowling teaches "using multiplexer to provide addresses not being translated, and to provide translated addresses for subsequent cycles" at page 11, lines 7 and 8 of the Official Action dated 12/26/2007. However, this multiplexer that Examiner suggest Dowling teaches is not what is claimed. Claim 7 claims "receiving a sequence of operand addresses from the multiplexer, and in response to the sequence of RFI operand addresses, translating the sequence of RFI operand addresses to form a sequence of different operand addresses in accordance with the translation pattern". In this manner, the multiplexer of claim 7 makes no selection to provide addresses not being translated or to provide translated addresses. Rather, the multiplexer of claim 7 is used for "selecting between the operand address from the instruction register for a first RFI operation and selecting the RFI update unit's output for subsequent RFI operations". In either case of RFI operations, the

operand addresses received from the multiplexer are translated. Dowling does not resolve the deficiencies of Pechanek 6,173,389 in this respect".

In response, Dowling discloses that in an FFT operation, the first operation does not need operand address translation, and the subsequent operand addresses can be translated (col. 7 lines 1-15). Dowling further discloses Fig 2, col. 8 line 52 to col. 9 line 15 that the multiplexer 203 select subsequent operands of FFT operations that required operand address translations (using address translation logic in 212), and also select first operand of FFT operations that does not required address translation (from 106, because it does not require operand address translation). Thus Dowling teaches the claim's limitation as being recited. Examiner uses the FFT operation as an example to show that Dowling clearly teaches the multiplex capable of selecting between the operand address from the instruction register for a first RFI operation and selecting the RFI update unit's output for subsequent RFI operations" as recited.

Regarding Appellant's argument "in either case of RFI operations, the operation addresses received from the multiplexer are translated".

Examiner notes that this comment appears referring to a different clause of the claim 7 which states "an address translation unit for accessing the memory in a translation pattern, receiving **a sequence** of operand addresses from the multiplexer and, in response to the sequence of RFI operand addresses, translating the sequence of RFI operand addresses to form a sequence of different operand addresses in accordance with the translation pattern, the different operand addresses each accessing a data element from the memory through the address input". In this clause, Appellant claim "**a sequence of operand addresses**" which can be reasonable viewed as any sequence of operand addresses. Thus Dowling's teaching operand addresses of subsequent

FFT operations met the claim's "a sequence of operand addresses". And Dowling clearly teaches that this sequence of operand addresses are all being translated to different operand addresses as claimed.

Therefore, Appellant's argument is not persuasive.

A6) Per claims 11-14, Appellant argues, "Claim 11 addresses, a "storage device located in the address translation memory device having data accessible at addressable locations, a second read address input internal to the address translation memory device for selecting data from the storage device during read operations, and a data output port". The Examiner admits that Dowling does not expressly teach the "claim's aspect of ports associating with the storage device" as claimed by claim 11 and relies upon Choquette in an effort to resolve this admitted deficiency. Choquette, however, **provides only ports external to the storage device** as described in Choquette paragraph 4. Thus, Choquette does not resolve the admitted deficiency of Dowling and this claim is not obvious"… " at page 22, lines 4-10 and lines 15-19 of the present specification. The storage unit 81 comprises **a first read address input port with address inputs 815**. The storage unit 810 also comprises a storage device 835 having **a second read address input port to the storage device 835**".

In response, Examiner notes that a **second** read address input port **to the storage device 835** is not found anywhere in the Fig 8A. Instead, Fig 8A clearly shows only single port to the storage device 835 which come from the same first read address input port with address input 815. In fact, the storage device 835 clearly shows as having one single read address input port A2' A1' A0'. Noted that the port A2' A1' A0' is external to storage device 835 and of course/inherently the port A2' A1' A0' is internal to the address translation memory device.

Appellant further argues, " One example is the block move operation described at page 23, line 15 - page 24, line 5. In order to move a block of four 32-bit registers in a processor without a swap instruction a "total of twelve 32-bit read operations and twelve 32-bit write operations would be required and would take at least 12-cycles to accomplish ....Using the techniques of the present invention, the whole operation can be accomplished in a single cycle and with no movement of the data ... thus demonstrating the effectiveness of the present techniques for low power and high performance". Such a capability is highly advantageous and cannot be accomplished by Dowling in combination with Choquette".

In response, Examiner notes that the block move feature is not recited in the claim. Additionally, Dowling teaches that the address translation memory device Fig 2 is a powerful programmable device readily to executing a sequence of operations by itself to carryout the sequence of addresses translation required in the FFT operation in an automatic and internally/by itself manner (Dowling's col. 8 line 62 to col. 9 line 15, "For example, the programmable AAU 212 can be programmed to provide automatic address indexing for FFT processing, Viterbi decoding etc).

A7) Per claim 15, Appellant argues, "Claim 15 was rejected under 35 U.S.C. § 103(a) as unpatentable over Dowling, Choquette as applied to claim 14 and further in view of Nair as discussed in the rejection of claim 3. Regarding claim 14, the Examiner did not apply Choquette in the rejection of claim 14. Rather, the Examiner rejected claim 14 based on the same rationale as of claim 2. As noted above, Intel, Choquette, and Nair do not resolve the admitted deficiencies of Dowling and the suggested rationale for rejecting claim 3 is not valid".

In response, claim 14 is a dependent of claim 11. Therefore claim 14 is rejected by Dowling , Choquette as applied in claim 11. Additionally, claim 14 recites the same features that Dowling teaches of claim 2. Thus the same reasons as discussed in the rejection of claim 2 applied in the rejection of claim 14.

Claim 15 is dependent of claim 14 which depend on claim 11. Therefore claim 15 is rejected by Dowling, Choquet as applied in claim 11. Additionally, claim 15 recites the same features that Nair teaches of claim 3. Thus claim 15 is further rejected by Nair with same reasons that Nair taught of claim 2. It would have been obvious to one of ordinary skill in the art at the time of invention to include bit-wise complementing function and matrix operations as suggested by Nair into Dowling's system modified by Choquette, thereby the system can further providing additional complex matrix operations in an automatic manner as shown in Nair's Table 1, and 2, and thereby the further improve the overall throughput in the system for executing matrix operations.

B) Appellant argues "..As shown above, the invention claimed is not suggested by the relied upon prior art. The references cited by the Examiner, if anything, teach away from the present invention. It is only in hindsight, after seeing the claimed invention, that the Examiner could combine the references as the Examiner has done..", "..In addition, the Examiner does not appear to have considered where the references diverge and teach away from the claimed invention…" "The Examiner's rejection suggests that the Examiner did not consider and appreciate the claims as a whole. The claims disclose a unique combination with many features and advantages not shown in the art. It appears that the Examiner has oversimplified the claims

and then searched the prior art for the constituent parts. Even with the claims as a guide, however, the Examiner did not recreate the claimed invention".

In response, Appellant does not point out any specifically error in the rejections of these claims with regard to the above allegations for the combinations of the recited references. Therefore, Examiner maintains the motivation of combinations of the recited references as stated in the rejections of the claims.

Additionally, Examiner presents details of the recited prior arts that teach/address all issues raised in the Appeal Brief, see section A.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/ Duc T. Doan/ DTD
Duc T. Doan
Examiner, Art Unit 2188

Conferees:

/Hyung S. Sough/ HSS
Supervisory Patent Examiner, Art Unit 2188
08/15/08

/Vincent F. Boccio/ VFB
Primary Examiner, Art Unit 2165
Appeal Specialist TC2100